Random Forest

Problem with decision trees: if the dataset changes slightly we could end up with a completely different tree.
Decision trees are highly sensitive to the training data → the model could fail to generalize.

Random Forest is a collection of multiple random decision trees much less sensitive to the training data.

① Build new datasets from original data — Bootstrapping: process of creating new data

id	x_0	x_1	x_2	x_3	x_4	y
0	1.3	1.9	1.1	1.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	1.7	6.7	4.2	5.3	4.8	1

id
2
0
2
4
5
5

id
2
1
3
1
4
4

id
4
1
3
0
0
2

id
3
3
2
5
1
1

The new dataset will contain the same number of rows of the original one
We perform random sampling with replacement (one row can occur multiple times)

② Train a decision tree on each of the bootstrapped datasets independently
We won't use all the features: instead we will select a subset of features for each tree

id
2
0
2
4
5
5

x_0, x_1

id
2
1
3
1
4
4

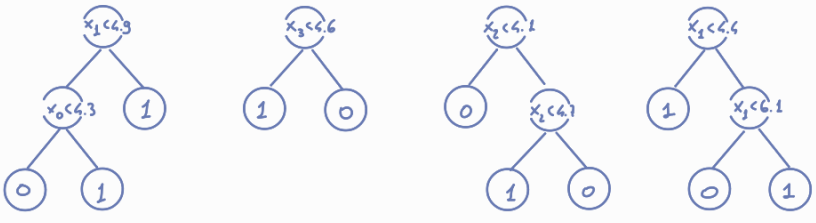
x_2, x_3

id
4
1
3
0
0
2

x_2, x_4

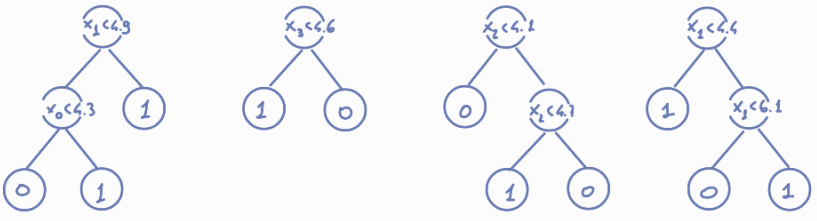
id
3
3
2
5
1
1

x_1, x_3 ← features used



③ In order to make a prediction, consider all the trees and take the majority voting

	x_0	x_1	x_2	x_3	x_4
new datapoint:	2.8	6.2	4.3	5.3	5.5



1 0 1 1 → majority voting: 1

Aggregation: combining results from multiple models

Bootstrapping + Aggregation = Bagging

2 random processes: ② bootstrapping and ② random feature selection

- bootstrapping → we don't use the same data for every tree; the final result is less sensitive to small variations in the original training data
- random feature selection → helps reducing the correlation between the variables

How many features to consider for each tree? close to the square root of the total number of features

- empirically proved $\sqrt{n}$ or $\log n$ of the total number of features